# FPGA Implementation of QPSK Modulator and Transmit Channelizer for Satellite Communication

Apoorva T R, Deepika Sankari G, Santhosh B

Department of Telecommunication, Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India

ABSTRACT

The main objective of this project is to design & implement a Verilog design of QPSK Digital Modulator based on the Xilinx FPGA. In recent years, Digital modulation has become less complex, more secure and more efficient in long distance transmission such as in satellite communication. Here, a 1Mbps QPSK modulator is designed for satellite communication applications. The input data generated using a PRBS generator and is then scrambled using a scrambler and a symbol mapper. The symbol mapper, maps the binary data to IQ symbols. A Root Raised Cosine filter is designed for interpolating the signals after which a CIC filter is used for interpolating the signals to the DAC sampling frequency. The DAC output is IQ modulated and up converted to an L-band RF signal (1.2GHz). Thus, helping in long distance transmission. A basic receiver is built to test the transmission characteristics. This consists of a Root Raised Cosine (RRC) decimator, a Zero-crossing detector and a descrambler.

A channelizer is designed at the transmitter which simultaneously up-converts a number of baseband signals to assemble a set of FDM channels in a single sampled data signal stream.

***KEYWORDS:*** *Quadrature Phase Shifting Keying (QPSK), Pseudo Random Binary Sequence (PRBS), Verilog, Field Programmable Gate Array (FPGA) and Channelizer*

***How to cite this paper:*** Apoorva T R | Deepika Sankari G | Santhosh B "FPGA Implementation of QPSK Modulator and Transmit Channelizer for Satellite Communication" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-4 | Issue-5, August 2020, pp.475-477, URL: www.ijtsrd.com/papers/ijtsrd31828.pdf

IJTSRD31828

## I. INTRODUCTION

One of the hottest topics that is undergoing immense research and development is satellite communication. Satellite communication has many applications one of which is its ability to communicate to long distances in a very less time. When, we talk about long distance communication many factors are to be taken in consideration, such as security, data rate, channel bandwidth, efficiency and many more. Modulation places a very important role in these, thus the type of modulation that is selected should well suited for one's requirements.

Quadrature Phase Shift keying, this form of modulation represents two bits as one and then it is transmitted. This helps in efficient usage of the available bandwidth. Since QPSK symbols are separated with a angle of π/2 fidelity is maintained thus offering power efficiency as well.

Here a QPSK modulator is built with a pseudo random sequence as the input which is then encoded using a scrambler and then to a QPSK symbol mapper after which the symbols are interpolated and passed through an RRC filter to obtain a pulse shaped signal. the pulse shaped signal is again filter with a CIC filter to match the sampling frequency of the hard ware.

The simulation of this modulator is done both in MATLAB and the RTL implementation is done in VIVADO. Using the modulator which was built a Channelizer for the same was implemented.

## II. QPSK MODULATOR DESIGN

This paper focuses on the design of a QPSK modulation and a transmit channelizer which are simulated on MATLAB for functionality and also implemented on the hardware using Verilog. The data to be modulated is in the form of a binary sequence. Here, a pseudo random binary sequence is considered. As the name suggests, the sequence is not completely random as it repeats itself after a certain period of time. The PRBS is later scrambled for the purpose of security and to avoid the problem of long sequences of 1's and 0's. Additive scrambler has the polynomial $1+z^{14}+z^{15}$. An Additive scrambler is used here which requires the initial state of both the scrambler and the descrambler to be the same. The synchronization word used here is of length 15 bits which is given by "10010101000000". The scrambled output is later given to the QPSK symbol generator which combines two consecutive bits into a symbol. A symbol is of the form I+jQ where I and Q represents the inphase and quadrature components. Thus, the possible bit combinations will have the following symbol representation:

TABLE I. QPSK Symbol Generation

| Input pair | In phase component | Quadrature component | Symbol representation |
|---|---|---|---|
| (0,0) | -1 | -1 | -1-j |
| (0,1) | -1 | 1 | -1+j |
| (1,0) | 1 | -1 | 1-j |
| (1,1) | 1 | 1 | 1+j |

Once the symbols are generated, they are mapped onto the plane with Inphase component along the X-axis and Quadrature component along the Y-axis. Hence there are four symbols present, one in each quadrant with a separation of 90$^0$. These symbols are later filtered using a Root Raised Cosine Interpolation filter. The filter has a roll-off factor of 0.25, with a span of 8 and containing 8 samples per symbol. The symbols are first interpolated by a factor of 8, in order to increase the number of samples and also to prevent Inter Symbol Interference (ISI). The next step would be to convolute the interpolated signal with the filter co-efficient. This will result in a pulse shaped signal which can be transmitted through a channel.

In order to verify, if the same input bits are received after the demodulation process, a basic receiver is designed. The first procedure at the receiver is the Root Raised Cosine Decimation. Here the first step is to convolute the pulse shaped signal with the same filter co-efficient which was used at the transmitter. After this process, the convolution overheads are filtered out and hence the number of samples will be same as that obtained after the interpolation process. The next step is the decimation process, where in the sampling rate is reduced by decimating by a factor of 8, which is the process of getting every eighth sample from the output. The output of the decimation process corresponds to the symbols that were interpolated. The de-mapping of these symbols will result in a binary sequence. This binary sequence is equivalent to the scrambled output. Thus, the binary sequence upon descrambling which uses the same synchronization word as the scrambler will give the input fed to the system which is the Pseudo Random Binary Sequence.

## III. CHANNELIZER DESIGN

Bandwidths are very expensive; hence the only wise choice is to use the available bandwidth to its full potential. This can be done by using a channelizer that lets us pass many signals at different frequencies using the same bandwidth that is available. For simulation, same signals that were generated using the QPSK modulator will be used. Two signals will be generated using two different symbol rates. After generation of two signals, one of the two signals will be multiplied by different sine waves so that they will have different frequencies. Here the aim is to create signals that will be centered at frequencies 400 KHz, 800 KHz,1200KHz and at 1600 KHz. The MATLAB simulation and the RTL synthesis is shown below in figures 2 and figure 5.

## IV. TESTBENCH

Test benches come in very handy as the input need not to be individually forced each time, we run a simulation. Thus, reducing time and man power required for every simulation that has to be run. The first step in writing a test bench is to define all the parameters with their respective sizes, that will be used includes clock, valid and the other inputs and the outputs of each module.

The initial values of reset and valid have to be defined. The reset is initialized keeping in mind in which condition the system goes into the reset condition, i.e. if it is active high then system goes under reset if the input is '1' or if it active low then the system goes into reset when the input is '0'. The clocks that are needed for each module to run is created usually using a clock divider which is by simply using a counter or a clock divider that is available in Vivado. After which instances for each module is created as required and connected using the wires which were predefined. It is very important that the size of the output of one module that is connected as the input to the other are same.

## V. HARDWARE IMPLEMENTATION

Here, a Xilinx ZC706 Evaluation Board is used which has a chip inside, ZynQ XC72045. ZynQ is a SoC (system on chip) which is, the system has both the FPGA and an ARM processor which is dual core. To the Xilinx evaluation board, the Analog Devices Evaluation board will be connected through the FMC port which has a chip named AD9361. This evaluation board is called the FM COMMS2. This is a RF Transmitter-Receiver which includes ADC, DAC, RF up-converter and so on. The FM COMMS2 consists of two RF transmitter-receiver pairs. Since this paper is based on a transmit channelizer, the cable is connected from the transmitter of one of the Tx-Rx pairs to the spectrum analyzer.

## VI. RESULTS

Simulation on MATLAB gives us the functionality of the system. The above process was simulated on MATLAB and the results are as follows.

In MATLAB, the symbols are processed as complex numbers with inphase component being the real part and quadrature component being the imaginary part. Upon QPSK modulation the following output is obtained.

In the output shown in the figure 1, the first figure indicates the PRBS sequence that is fed as the input. The second figure represents the Root Raised Cosine filter coefficients. The third figure represents the pulse shaped signal and the last one represents the data recovered.

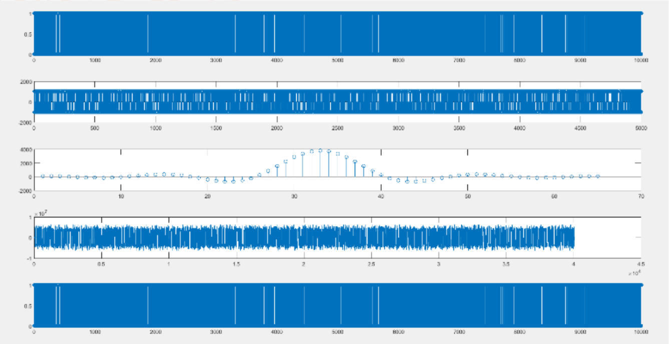


**Fig.1. QPSK Modulation and Verification in MATLAB**

The pulse shaped signal after being passed through the CIC filter, as seen in the figure 2, the signal at the center is at a symbol rate of 100kHz and the remaining four signals are at a symbol rate of 200kHz which are multiplied with sine waves of frequencies 400kHz,800kHz,1200kHz and 1600kHz respectively and added. This is done in order to transmit all the five signals through the same channel with a frequency separation of 400kHz.

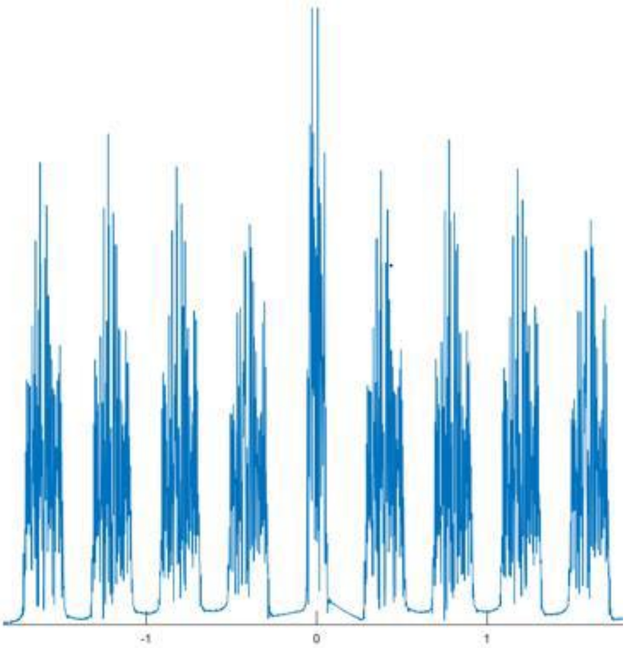
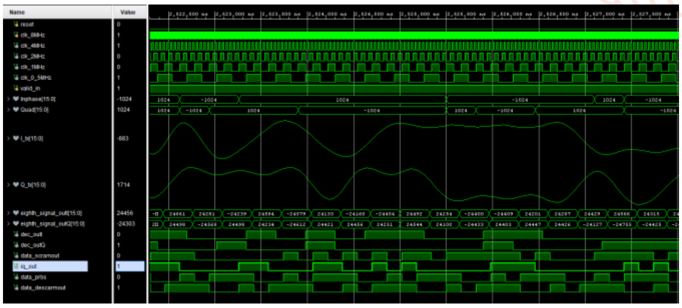


**Fig.2. Transmit Channelizer output**

In case of Verilog, the QPSK modulation is performed and the figure 3, shows the pulse shaped signal and the verification of the process. The output shows the following: reset, clocks at different frequencies, valid input, Inphase and Quadrature components of a symbol, and the pulse shaped signal truncated to 16 bits (I_tx and R_tx),decimated inphase and quadrature components, bits recovered from the decimated signal(I and Q bits), scrambled data, combined I and Q bits, input PRBS and descrambled output in order.

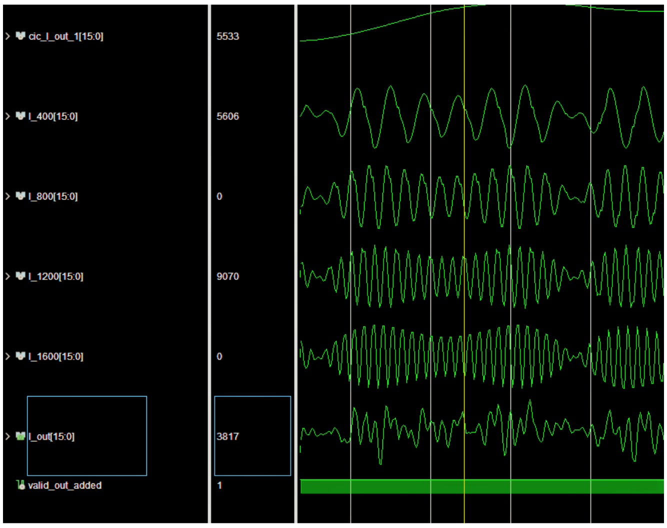
In case of Verilog, the sine waves are generated using the coefficients obtained in MATLAB and multiplied with the pulse shaped signal having a symbol rate of 200kHz. For the signal with symbol rate of 100kHz the output of the CIC filter serves as the output. In case of Verilog as the inphase and quadrature components are dealt with separately, the respective I and Q components of the five signals are added separately.



| Name | Value |
|---|---|
| reset | 0 |
| clk_8MHz | 1 |
| clk_4MHz | 1 |
| clk_2MHz | 0 |
| clk_1MHz | 0 |
| clk_0_5MHz | 1 |
| valid_in | 1 |
| Inphase[15:0] | -1024 |
| Quad[15:0] | 1024 |
| I_tx[15:0] | -883 |
| Q_tx[15:0] | 1714 |
| eighth_signal_outI[15:0] | 24456 |
| eighth_signal_outQ[15:0] | -24303 |
| dec_outI | 0 |
| dec_outQ | 1 |
| data_scramout | 0 |
| iq_out | 1 |
| data_prbs | 0 |
| data_descarmout | 1 |

**Fig.3. QPSK Modulation and Verification in VERILOG**

The figure 4 represents the inphase components of the five signals. As these signals are obtained after multiplying with a sine wave, the signal has an envelope of a sine wave. The quadrature components are also processed in the same way as the inphase components and hence it looks similar to the figure 4. Later, these inphase and quadrature components serve as the input to the hardware.



| | |
|---|---|
| cic_I_out_1[15:0] | 5533 |
| I_400[15:0] | 5606 |
| I_800[15:0] | 0 |
| I_1200[15:0] | 9070 |
| I_1600[15:0] | 0 |
| I_out[15:0] | 3817 |
| valid_out_added | 1 |

**Figure 4: Channelizer**

The hardware implementation of the above gives the following results.

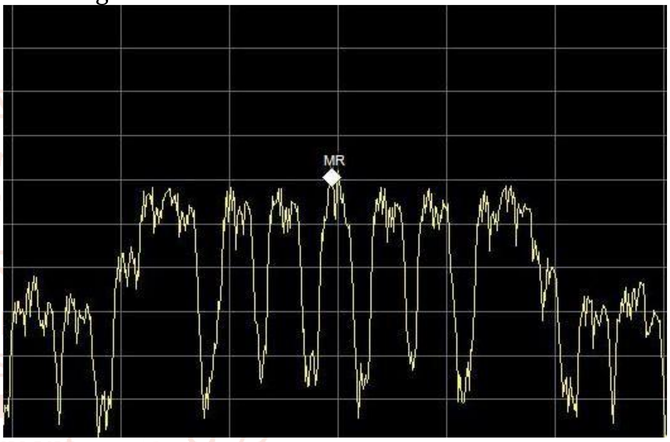


**Figure 5: Spectrum analyzer output**

## REFERENCES

[1] Amean Al safi and Bradley Bazuin, "FPGA Based Implementation of BPSK and QPSK Modulators using Address Reverse Accumulators", IEEE,2016.

[2] Akhil Khanna, Anju and Harsh Jain" Design and Synthesis of Bandwidth Efficient QPSK Modulator for Low Power VLSI Design", Jaiswal ICECS 2015.

[3] K. Mounica, S. Mohan Das P. Uday Kumar." A Verilog Design in FPGA Implementation of Quadrature Phase Shift Keying (QPSK) Digital Modulator", International journal of engineering sciences and research technology, 2013.

[4] Vishakha Pareek School "A Novel Implementation of QPSK Modulator on FPGA", 2016.

[5] Wenmiao Song & Qiongqiong Yao. "Design and Implement of QPSK Modem Based on FPGA" IEEE, 2010.

[6] Digital Design, an embedded systems Approach using - Peter J. Ashenden.

[7] Digital Signal Processing, 5th edition - by V. K. Ingle and J. G. *Proakis.*

[8] MATLAB and its applications in Engineering by Raj Kumar Bansal, Ashok Kumar Goel, Manoj Kumar Sharma.